Search Results

Search Results for: **[java and just in time and krall]**
Found **35** of **127,132 searched.**

Search within Results

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 20 of 35 **short listing**

Prev Page 1 2 Next Page

1 Techniques for obtaining high performance in Java programs 88%
Iffat H. Kazi , Howard H. Chen , Berdenia Stanley , David J. Lilja
ACM Computing Surveys (CSUR) September 2000
Volume 32 Issue 3
This survey describes research directions in techniques to improve the performance of programs written in the Java programming language. The standard technique for Java execution is interpretation, which provides for extensive portability of programs. A Java interpreter dynamically executes Java bytecodes, which comprise the instruction set of the Java Virtual Machine (JVM). Execution time performance of Java programs can be improved through compilation, possibly at the expense of portabili ...

2 A brief history of just-in-time 87%
John Aycock
ACM Computing Surveys (CSUR) June 2003
Volume 35 Issue 2
Software systems have been using "just-in-time" compilation (JIT) techniques since the 1960s. Broadly, JIT compilation includes any translation performed dynamically, after a program has started execution. We examine the motivation behind JIT compilation and constraints imposed on JIT compilation systems, and present a classification scheme for such systems. This classification emerges as we survey forty years of JIT work, from 1960--2000.

3 Efficient Java exception handling in just-in-time compilation 85%
Seungil Lee , Byung-Sun Yang , Suhyun Kim , Seongbae Park , Soo-Mook Moon , Kemal Ebcioğlu , Erik Altman
Pr ceedings f the ACM 2000 c nference n Java Grande June 2000

Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: L24 and java

Display: 100 **Documents in Display Format:** TI,AB **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search | Clear | Interrupt

Search History

DATE: Friday, February 13, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L25	L24 and java	1	L25
L24	(revert$ near5 instruction$)	140	L24
L23	l4 and (rever$ or oppos$)	0	L23
L22	l4 and (memory$ or data or cache$)	1	L22
L21	l4 and execut$	1	L21
L20	l4 and (load$ OR STOR$)	1	L20
L19	l4 and load$ OR STOR$	1083749	L19
L18	l4 and (except$ or error$ or signal$)	0	L18
L17	l4 and float$	0	L17
L16	l4 and double	0	L16
L15	l4 and swap$	0	L15
L14	l4 and (operand or depth$ or stack$)	1	L14
L13	(sipush or bipush) and (overflow$ or underflow$)	17	L13
L12	(sipush or bipush) same (overflow$ or underflow$)	0	L12
L11	(sipush or bipush) near5 (overflow$ or underflow$)	0	L11
L10	l1 and sipush and bipush	0	L10

L9	l4 and (overflow$ or underflow$)	0	L9
L8	l4 and overflow$ and underflow$	0	L8
DB=USOC; PLUR=YES; OP=ADJ			
L7	l4 and overflow$ and underflow$	0	L7
DB=USPT,USOC; PLUR=YES; OP=ADJ			
L6	l4 and overflow$ and underflow$	0	L6
L5	l1 and overflow$ and underflow$	1	L5
DB=USPT; PLUR=YES; OP=ADJ			
L4	5875336.pn.	1	L4
L3	l1 and execut$	1	L3
L2	L1 and (execut$ near5 stack$)	1	L2
L1	6332215.pn.	1	L1

END OF SEARCH HISTORY

Search Results

Search Results for: **[processor and translate and register and instruction and stack and java]**
Found **200** of **127,132 searched.**

Search within Results

Results 1 - 20 of 200 **short listing**

Prev Page **1 2 3 4 5 6 7 8 9 10** Next Page

1 Techniques for obtaining high performance in Java programs 99%
Iffat H. Kazi , Howard H. Chen , Berdenia Stanley , David J. Lilja
ACM Computing Surveys (CSUR) September 2000
Volume 32 Issue 3
This survey describes research directions in techniques to improve the performance of programs written in the Java programming language. The standard technique for Java execution is interpretation, which provides for extensive portability of programs. A Java interpreter dynamically executes Java bytecodes, which comprise the instruction set of the Java Virtual Machine (JVM). Execution time performance of Java programs can be improved through compilation, possibly at the expense of portabili ...

2 Improving Java performance using hardware translation 94%
Ramesh Radhakrishnan , Ravi Bhargava , Lizy K. John
Proceedings of the 15th international conference on Supercomputing June 2001
State of the art Java Virtual Machines with Just-In-Time (JIT) compilers make use of advanced compiler techniques, run-time profiling and adaptive compilation to improve performance. However, these techniques for alleviating performance bottlenecks are more effective in long running workloads, such as server applications. Short running Java programs, or client workloads, spend a large fraction of their execution time in compilation instead of useful execution when run using JIT compilers. In ...

3 LLVA: A Low-level Virtual Instruction Set Architecture 92%
Vikram Adve , Chris Lattner , Michael Brukman , Anand Shukla , Brian Gaeke
Pr ceedings f the 36th Annual IEEE/ACM Internati nal Symp sium n Micr architecture December 2003
A virtual instruction set architecture (V-ISA) implementedvia a processor-specific

1) Enter a single keyword, phrase, or Boolean expression. Example: acoustic imaging (means the phrase acoustic imaging plus any stem variations)
2) Limit your search by using search operators and field codes, if desired.
Example: optical <and> (fiber <or> fibre) <in> ti
3) Limit the results by selecting Search Options.
4) Click Search. See Search Examples

```
cpu and stack and instruction
and register and overflow and
underflow and java
```

Start Search Clear

Note: This function returns plural and suffixed forms of the keyword(s).

Search operators: <and> <or> <not> <in> More

Field codes: au (author), ti (title), ab (abstract), jn (publication name), de (index term) More

Search Options:

Select publication types:

- ☑ IEEE Journals
- ☑ IEE Journals
- ☑ IEEE Conference proceedings
- ☑ IEE Conference proceedings
- ☑ IEEE Standards

Select years to search:

From year: All to Present

Organize search results by:

Sort by: Relevance
In: Descending order

List 15 Results per page

1) Enter a single keyword, phrase, or Boolean expression. Example: acoustic imaging (means the phrase acoustic imaging plus any stem variations)
2) Limit your search by using search operators and field codes, if desired.
Example: optical <and> (fiber <or> fibre) <in> ti
3) Limit the results by selecting Search Options.
4) Click Search. See Search Examples

```
cpu and stack and instruction
and register and translate
```

Start Search Clear

Note: This function returns plural and suffixed forms of the keyword(s).

Search operators: <and> <or> <not> <in> More

Field codes: au (author), ti (title), ab (abstract), jn (publication name), de (index term) More

Search Options:

Select publication types:

- [x] IEEE Journals
- [x] IEE Journals
- [x] IEEE Conference proceedings
- [x] IEE Conference proceedings
- [x] IEEE Standards

Select years to search:

From year: All to Present

Organize search results by:

Sort by: Relevance
In: Descending order

List 15 Results per page

Your search matched **2** of **1003743** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

processor and stack and instruction and register and Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Early load address resolution via register tracking**
Bekeman, M.; Yoaz, A.; Gabbay, F.; Jourdan, S.; Kalaev, M.; Ronen, R.;
Computer Architecture, 2000. Proceedings of the 27th International Symposiu on , 10-14 June 2000
Pages:306 - 315

[Abstract] [PDF Full-Text (964 KB)] **IEEE CNF**

---

2 **An X86 microprocessor with multimedia extensions**
Draper, D.A.; Crowley, M.P.; Holst, J.; Favor, G.; Schoy, A.; Ben-Meir, A.; Tru Khanna, R.; Wendell, D.; Krishna, R.; Nolan, J.; Partovi, H.; Johnson, M.; Lee Mallick, D.; Frydel, G.; Vuong, A.; Yu, S.; Maley, R.; Kauffmann, B.;
Solid-State Circuits Conference, 1997. Digest of Technical Papers. 44th ISSCC 1997 IEEE International , 6-8 Feb. 1997
Pages:172 - 173, 450

[Abstract] [PDF Full-Text (872 KB)] **IEEE CNF**

Refine Search

Search Results -

Terms	Documents
(717/150 |717/151 |717/152 |717/153).ccls.	325

Database:
US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Search: L13

Refine Search | Recall Text | Clear | Interrupt

Search History

DATE: Sunday, February 15, 2004 Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
DB=USPT; PLUR=YES; OP=ADJ			
L13	717/150,151,152,153.ccls.	325	L13
L12	717/146,147,148,149.ccls.	411	L12
L11	717/136,137,138,139,140.ccls.	494	L11
DB=TDBD; PLUR=YES; OP=ADJ			
L10	L8	0	L10
DB=DWPI; PLUR=YES; OP=ADJ			
L9	L8	0	L9
DB=JPAB; PLUR=YES; OP=ADJ			
L8	cpu and (register$ near4 instruction$) and (stack$ near5 instruction$) and (translat$ or decod$ or compil$)and overflow$ and underflow$	0	L8
DB=EPAB; PLUR=YES; OP=ADJ			
L7	cpu and (register$ near4 instruction$) and (stack$ near5 instruction$) and (translat$ or decod$ or compil$)and overflow$ and underflow$	0	L7

DB=USOC; PLUR=YES; OP=ADJ			
L6	cpu and (register$ near4 instruction$) and (stack$ near5 instruction$) and (translat$ or decod$ or compil$)and overflow$ and underflow$	2	L6
DB=PGPB; PLUR=YES; OP=ADJ			
L5	cpu and (register$ near4 instruction$) and (stack$ near5 instruction$) and (translat$ or decod$ or compil$)and overflow$ and underflow$	14	L5
DB=USPT; PLUR=YES; OP=ADJ			
L4	L3 and java	17	L4
L3	L2 and exception$	162	L3
L2	L1 and overflow$ and underflow$	204	L2
L1	cpu and (register$ near4 instruction$) and (stack$ near5 instruction$) and (translat$ or decod$ or compil$)	941	L1

END OF SEARCH HISTORY